PTOLEMAIC INDEXING

*Magnus Lie Hetland**

Abstract. This paper discusses a new family of bounds for use in similarity search, related to those used in metric indexing, but based on Ptolemy's inequality, rather than the metric axioms. Ptolemy's inequality holds for the well-known Euclidean distance, but is also shown here to hold for quadratic form metrics in general, with Mahalanobis distance as an important special case. The inequality is examined empirically on both synthetic and real-world data sets and is also found to hold approximately, with a very low degree of error, for important distances such as the angular pseudometric and several L_p norms. Indexing experiments demonstrate a highly increased filtering power compared to existing, triangular methods. It is also shown that combining the Ptolemaic and triangular filtering can lead to better results than using either approach on its own.

1 Introduction

In similarity search, data objects are retrieved based on their similarity to a query object; as for other modes of information retrieval, the related indexing methods seek to improve the efficiency the search. Two approaches seem to dominate the field: spatial access methods [1–3], based on coordinate geometry, and metric access methods [4–7], based on the metric axioms. Similarity retrieval with spatial access methods is often restricted to L_p norms, or other norms with predictable behavior in $\mathbb{R}^k$, while metric access methods are designed to work with a broader class of distances—basically any distance that satisfies the triangular inequality. This gives the metric approach a wider field of application, by forgoing some assumptions about the data, which in some cases results in lower performance. Interestingly, even in cases where spatial access methods are applicable, metric indexing may be superior in dealing with high-dimensional data, because of its ability to bypass the so-called representational dimensionality and deal with the intrinsic dimensionality of the data directly [see, e.g., 8].

It seems clear that there are advantages both to making strong assumptions about the data, as in the spatial approach, and in working directly with the distances, as in the metric approach. The direction taken in this paper is to apply a new set of restrictions to the distance, separate from the metric axioms, still without making any kind of coordinate-based assumptions. The method that is introduced is shown to have a potential for highly increased filtering power, and while it holds for the square root of any metric, it is also shown analytically to apply to quadratic form metrics in general (with Euclidean distance as an important special case), and empirically, as an approximation with a very low degree

*Dept. of Computer Science, Norwegian University of Science and Technology, mlh@idi.ntnu.no

of error, to important cases such as the angular pseudometric, edit distance between strings, and several L_p norms.

This paper is a revised version of a preprint from 2009 [7]. While I have edited the paper for clarity, and at times adjusted the emphasis on various topics, the substance remains the same. Since the original was published, a few dozen papers have appeared that discuss or build on its ideas, and I refer to a couple of the more relevant of these publications throughout. The main contributions of the paper may be summed up as follows.

(*i*) It gives an example of indexing that goes beyond the established metric axioms, and demonstrates that doing so can have merit, poining the way to a broader field of exploration.

(*ii*) It introduces a filtering condition where the number of available bounds grows quadratically with the number of sample objects, and thus with the amount of space and the number of distance computations needed to perform the filtering. It also shows that this increase leads directly to an increase in filtering power. This introduces a tradeoff between the number of bounds and the number of false positives—a tradeoff that depends on the cost of the distance function.

(*iii*) It shows how to take the bounds beyond basic pivot filtering and apply them to more general index structures, potentially permitting more realistic, large-scale retrieval systems to be built.

(*iv*) It shows the equivalence of quadratic form metrics and Ptolemaic norm metrics (see Theorem 1).

The first point has already led others to look for axiom sets tailored to specific data, for example [c.f., 9, 10]. The second point has led to indexing methods for the highly expensive *signature quadratic form distance*—methods that clearly beat the competing metric indexes, as measured in actual search time [11]. The third point has been built upon to construct fully formed, competetive index structures [12]. Finally, the equivalence theorem has laid the foundations for the proof that the signature quadratic form distance is a Ptolemaic metric [11, 12].

This, then, is not primarily a paper on algorithm engineering; I do not attempt to fully design specific index structures, with specific heuristics for index traversal, or to perform exhaustive comparative experiments; these are directions of research that have come to fruition elsewhere. Here, I simply aim to present the basic idea of Ptolemaic indexing, and to demonstrate that it holds promise for wresting additional filtering power from a given set of sample objects. And though I touch upon spaces that are only partially Ptolemaic, in the interest of mapping out the territory, my focus here is primarily on increased filtering power for exact search.

2 Basic Concepts

The *similarity* in similarity retrieval is usually formalized, inversely, using a *dissimilarity function*, a nonnegative real-valued function $d(\cdot\,,\cdot)$ over some universe of objects, $\mathbb{U}$. For typographic convenience, I will generally follow the old-fashioned convention[1] of abbreviating $d(x, y)$ as xy. A distance query consists of a query object q, and some form of distance threshold, either given as a range (radius) r, or as a neighbor count k. For the range query, all objects o for which $qo \leq r$ are returned; for the k-nearest-neighbor query (kNN), the k nearest neighbors of q are returned, with ties broken arbitrarily. The kNN queries can be implemented in terms of range queries in a quite general manner [5].

For a function to qualify as a dissimilarity function (or *premetric*) the value of xx must be zero. It is generally assumed (mainly for convenience) that the dissimilarity is *symmetric* ($xy = yx$, giving us a *distance*) and *isolating* ($xy = 0 \Leftrightarrow x = y$, yielding a *semimetric*). From a metric indexing perspective, it is most crucial that the distance be *triangular* (obeying the triangular inequality, $xz \leq xy + yz$). A *metric* is any symmetric, isolating, triangular dissimilarity function. Metrics are related to the concept of *norms*: A *norm metric* is a metric of the form $xy = \|x - y\|$, where x and y are vectors and $\|\cdot\|$ is a norm.

Many distances satisfy the metric properties, including several important distances between sets, strings and vectors, and metric indexing is the main approach in distance-based retrieval. The metricity, and in particular triangularity, is exploited to construct lower bounds for efficient filtering and partitioning. As discussed in an earlier tutorial [7], there are several ways of using the triangular inequality in metric indexing, usually by pre-computing the distances between certain sample objects (called *pivots* or *centers*, depending on their use) and the other objects in the data set. By computing the distances between the query and the same sample objects (or some of them), triangularity can be used to filter out objects that clearly do not satisfy the criteria of the query.

Even though most distance-based indexing has centered on the metric axioms, this paper focuses on another property, known as *Ptolemy's inequality*, which states that

$$xv \cdot yu \leq xy \cdot uv + xu \cdot yv\,, \tag{1}$$

for all objects x, y, u, v. Premetrics satisfying this inequality are called *Ptolemaic*.[2] In the terms of the Euclidean plane: For any quadrilateral, the sum of the pairwise products of opposing sides is greater than or equal to the product of the diagonals (see Fig. 1).

It is a well-known fact that Euclidean distance is Ptolemaic [15]. There are many proofs for this, some quite involved, but it can be shown quite simply, using the idea of *inversion* [16]. A sketch of the argument is given in Fig. 2. What is perhaps less well known is that there is a natural connection between Ptolemaic metrics on vector spaces and a generalization of Euclidean distance—a family of distances collectively referred to as *quadratic form distance*. A quadratic form distance may be expressed as

[1]See, for example, Wilson [13].

[2]Note that Ptolemy's inequality neither implies nor is implied by triangularity in general [14].

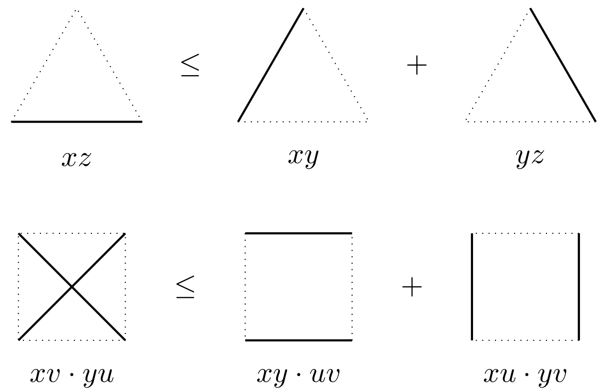


Figure 1: An illustration of the triangle inequality (top) and Ptolemy's inequality (bottom).

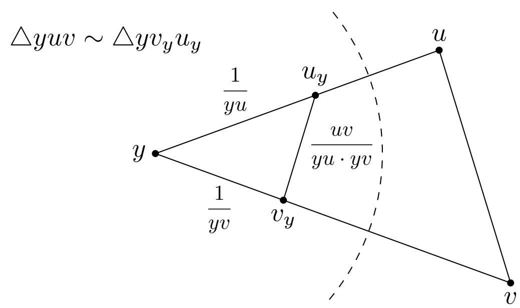


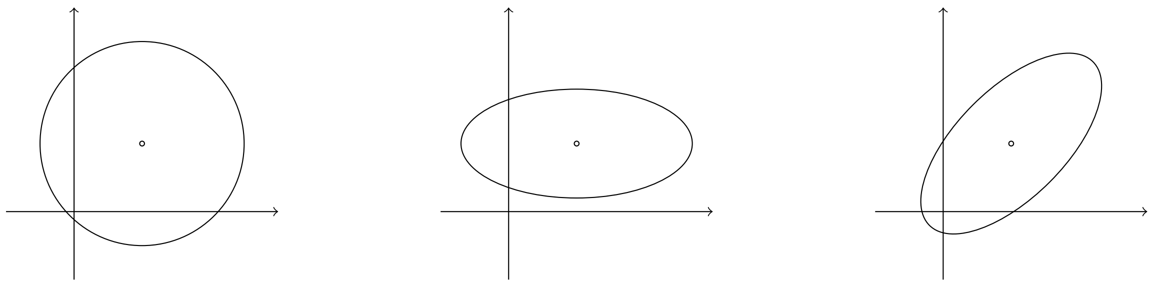
Figure 2: Getting Ptolemy's inequality from the triangular inequality through inversion in Euclidean space: The points are inverted with respect to y, with an inversion radius of unity (dashed). Adding a third point, x, and applying the triangular inequality to u_y, v_y and x_y yields Ptolemy's inequality for x, y, u and v.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}(x_i - y_i)(x_j - y_j)}\,,$$

or, in matrix notation, $\sqrt{\mathbf{z}'\mathbf{A}\mathbf{z}}$, where $\mathbf{x}$ and $\mathbf{y}$ are vectors, and $\mathbf{z} = \mathbf{x} - \mathbf{y}$. The weight matrix $\mathbf{A} = [a_{ij}]$ is a measure of "unrelatedness" between the dimensions, which uniquely defines the distance. We can, without loss of generality, assume that $\mathbf{A}$ is symmetric, as any antisymmetries will have no bearing on the distance [17]. In order for the distance to be a metric, $\mathbf{A}$ must also be positive-definite.[3]

Quadratic form distances take into account possible correlations between the dimensions of the vector space, which makes them especially suited for comparing histograms [see,

[3]Some sources require only positive-semidefiniteness [e.g., 6], but this would result in a pseudometric, allowing a distance of zero between different objects. It is possible to relax the requirement on $\mathbf{A}$ by adding requirements to the inputs [17].



(a) Euclidean (b) Weighted Euclidean (c) Quadratic form

Figure 3: Metric balls for three related distance types in $\mathbb{R}^2$.

e.g., 18]. For example, when comparing color histograms, it might be natural to compare the bin (that is, dimension) for orange to those of red and yellow [17]. If $\mathbf{A}$ is restricted to a diagonal matrix, a weighted Euclidean distance results, with the identity matrix leading to ordinary Euclidean distance (see Fig. 3). An important kind of quadratic form distance is the Mahalanobis distance, where $\mathbf{A}$ is normally set to the inverse of the covariance matrix of a data sample.

The fact that this important family of distances is usable with the techniques presented in this paper is expressed in the following theorem.

Theorem 1. *A distance function on $\mathbb{R}^n$ is a quadratic form metric if and only if it is a Ptolemaic norm metric.*

Proof. Let the quadratic form metric $d(\mathbf{x}, \mathbf{y})$ be $\sqrt{\mathbf{z}'\mathbf{A}\mathbf{z}}$, where $\mathbf{z} = \mathbf{x} - \mathbf{y}$. Because $\mathbf{A}$ is symmetric positive-definite, $\mathbf{z}'\mathbf{A}\mathbf{z}$ defines an inner product, making any such d a norm metric based on an inner product norm (a norm of the form $\|\mathbf{z}\| = \sqrt{\langle \mathbf{z}, \mathbf{z} \rangle}$, where $\langle \cdot, \cdot \rangle$ is an inner product). Note that the converse also holds: Any inner product norm distance on $\mathbb{R}^n$ can be expressed as a quadratic form metric (with a positive-definite $\mathbf{A}$). It is known that a norm metric is Ptolemaic if and only if its norm is an inner product norm [15, 19], which gives us the theorem. $\square$

As pointed out by Skopal et al. [20], if $\mathbf{A}$ is static, a distance-preserving transform into Euclidean space can be used to eliminate the quadratic form from the search. This is not possible, however, if $\mathbf{A}$ is constructed on demand, as in the *signature* quadratic form distance, for example [21]. Computing such distances will tend to be expensive, making them prime candidates for Ptolemaic indexing.

Beyond quadratic forms, there are certainly many other Ptolemaic metrics (with the discrete metric, $d(x, y) = 1 \Leftrightarrow x \neq y$, as an obvious example, and the chordal metric on the unit Riemann sphere as a possibly less obvious one [22]). In fact, for any metric d, the metric $\sqrt{d(\cdot)}$ is Ptolemaic [23]. In terms of distance orderings, and therefore similarity queries, this new metric is equivalent to the original, meaning that the techniques in this paper are applicable to all metrics. However, the transform will increase its intrinsic dimensionality [24], generally making the process of indexing harder. As shown in

Sect. 6.2, several non-Ptolemaic metrics may be "sufficiently Ptolemaic" to be used with the Ptolemaic indexing techniques *without* such a transform, although only for approximate queries.

3 Related Work

As discussed at length elsewhere [3–7], there are many published methods that deal with indexing distances based on the metric properties. While there has been an increasing focus on reducing I/O and general CPU time, the primary aim of most publications has been minimizing the number of distance computations, based on the assumption that the distance is highly expensive to compute.[4] While focusing exclusively on this one performance criterion may not be altogether realistic, yielding methods with linear query time [26, 27] or quadratic memory use [28, 29], for example, it has proven a useful foundation on which methods with more nuanced performance properties could be built [e.g., 30–33]. This paper also focuses on minimizing distance computations, setting aside related questions of algorithm engineering for later.

The main mechanism through which distance calculations may be avoided is through various forms of *filtering* or *exclusion*, using lower bounds [7].[5] If the structure of the data objects is known, then very precise, yet cheap, lower bounds can be constructed [34], but for metric indexing, only the general properties of the distance may be used. As described in Sect. 2, this is done by storing precomputed distances, or ranges of distances, involving the data set and certain sample objects (*centers* or *pivots*).[6]

Rather than listing existing indexing methods, only the most relevant of the basic principles will be addressed here. Two rather general theorems contain the majority of the indexing principles as special cases. The first of these, dealing with metric balls, is given here without proof. The second, dealing with generalized hyperplanes, as well as more details and proofs for both theorems can be found in the aforementioned tutorial [7], the survey by Hjaltason and Samet [5] or the textbook by Zezula et al. [6]. In the following, let o, p and q be objects in a universe $\mathbb{U}$, and let the implicit distance d be a metric over $\mathbb{U}$.

Theorem 2. *The value of qo may be bounded as*

$$\max\{op^- - qp^+, qp^- - op^+\} \le qo \le qp^+ + op^+ ,$$

where $uv^- \le uv \le uv^+$, for any objects u, v in $\mathbb{U}$. □

The expressions uv^- and uv^+ refer to known lower and upper bounds for uv, as, in some cases, the exact distances may not be known. For example, p may be the center of a

[4]This assumption may very well stem from the seminal work of Feustel and Shapiro [25], where calculation of the metric involved comparing every permutation of the node sets of two graphs.

[5]The converse, *inclusion* using upper bounds, is also possible, but less frequently useful, simply because most of the data should normally be excluded from the result set.

[6]In the case of generalized hyperplane indexing, the information stored is simply which center is closest to a given object, and this is implicit in the structure.

metric ball containing o, with covering radius r. In that case $po^- = 0$ and $po^+ = r$. If the query–pivot distance is known, we get the lower bound

$$qp - r \leq qo\,, \tag{2}$$

which is exactly the bound used to check for overlap between a query ball and a bounding ball in a metric tree, for example.

4 Ptolemaic Pivot Filtering

In a manner similar to Theorem 2, Ptolemy's inequality may be used to construct lower bounds for filtering. In the following, the technique known as *pivoting* is used. The derivation of a more general bound is deferred to Sect. 5. Triangular pivoting is based on the following lower bound, a special case of the one in Theorem 2, where the distances are known exactly:

$$qo \geq |qp - op| \tag{3}$$

Here, q is the query object, o is a candidate result object, while p is a so-called *pivot* object, whose function is to help construct the bound. This bound is sometimes known as the *inverse* triangular inequality, and follows from basic restructuring of the original:

$$\begin{aligned} op + qo &\geq qp \\ qo &\geq qp - op\,, \end{aligned} \tag{4}$$

and, in the same manner,

$$qo \geq op - qp\,. \tag{5}$$

Together, (4) and (5) lead directly to (3). The bound is normally strengthened by using a set of several pivots, P:

$$qo \geq \max_{p \in P} |qp - op|$$

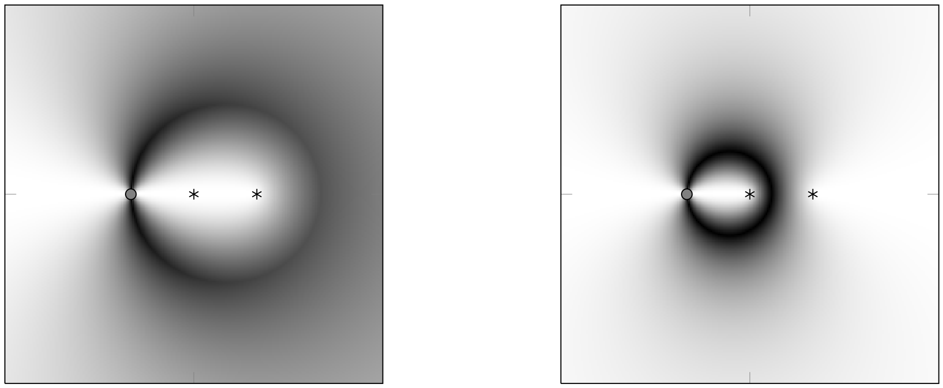
A similar derivation can be made for Ptolemaic distances. In the following, q, p, and o retain their previous meaning, but we also add another pivot object, s:

$$\begin{aligned} qs \cdot op + qo \cdot ps &\geq qp \cdot os \\ qo \cdot ps &\geq qp \cdot os - qs \cdot op \\ qo &\geq (qp \cdot os - qs \cdot op)/ps \end{aligned} \tag{6}$$

Here we can maximize over all pairs of pivots:

$$qo \geq \max_{p,s \in P} \frac{qp \cdot os - qs \cdot op}{ps} \tag{7}$$

By exchanging p and s in (6), we could exchange the terms in the numerator, allowing us to use the absolute value in (7). This would not strengthen the bound as it stands, but would allow us halve the number of pivot pairs examined.



(a) Triangular pivoting (b) Ptolemaic pivoting

Figure 4: Accuracy, measured as the ratio between the pivoting bound and the actual distance, from 0% (black) to 100% (white). The three points represent, from left to right, a query and two pivots.

For the normal pivoting bound to be useful, the pivot should be closer to either the query or to the candidate object; the difference in the two distances is what gives the bound its filtering power. For the Ptolemaic pivoting bound, it seems that one way of getting good results would be to have one pivot close to the query, while the other is close to the candidate object, giving a high value for the numerator in (7). However, this intuition is tempered by the denominator, which dictates that the pivots should also be close to each other. Invariably, the tradeoff here will need to be based on empirical considerations. As long as the distance matrix between the pivots is precomputed, the bound can be computed for every pair of pivots, and the maximum used as the final bound. As will be shown in Sect. 6, this Ptolemaic pivoting bound is a significant improvement over the classical triangular one.

The difference between triangular and Ptolemaic pivoting in the Euclidean plane is illustrated in Fig. 4, where the ratio between bound and distance from a query at $(-1, 0)$, with the pivots placed at $(0, 0)$ and $(1, 0)$, is plotted for each point, where zero (a non-informative lower bound) is black and one (a perfect bound) is white. While the the bound in (7) is the one examined further in this paper, the ideas of Ptolemaic indexing have wider implications. In the following section, a more general theorem (Theorem 3) is given, which is a Ptolemaic analogue of Theorem 2.

5 Generalizing the Ptolemaic Bound

The following theorem generalizes the bound (7), as a Ptolemaic analogue of Theorem 2. This generalization is included as a starting point for new indexing methods, and is not examined empirically in this paper.

Theorem 3. *Let o, p, q and s be objects in a universe $\mathbb{U}$, and let d be a Ptolemaic distance over $\mathbb{U}$. The value of qo may then be bounded as*

$$\frac{1}{ps^+} \cdot \max \left\{ \begin{array}{l} qs^- \cdot op^- - qp^+ \cdot os^+ , \\ qp^- \cdot os^- - qs^+ \cdot op^+ \end{array} \right\} \leq qo \leq \frac{1}{ps^-} \cdot \left(qp^+ \cdot os^+ + qs^+ \cdot op^+ \right) ,$$

where $uv^- \leq uv \leq uv^+$, for any objects u, v in $\mathbb{U}$.

Proof. The two cases of the lower bound correspond to the two possible orderings of the products in the numerator of the lower bound (7), both of which are permissible. The only change here is that instead of the exact values, we use upper and lower limits. The lower limits occur before the subtractions, while the upper limits occur after, as well as in the denominator. Given that all distances are non-negative, the lower (resp. upper) bounds also provide lower (resp. upper) bounds on the products. Therefore, these substitutions can only lower the value of the bound, and hence the inequality still holds. The upper bound follows from the Ptolemaic inequality

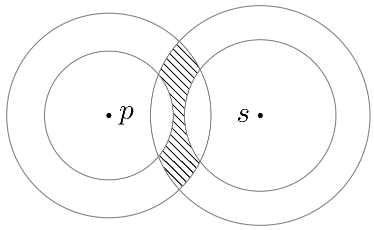
$$ps \cdot qo \leq qp \cdot os + qs \cdot op .$$

Dividing by ps and safely substituting upper limits in the numerator and a lower limit in the denominator, we arrive at the upper bound. □

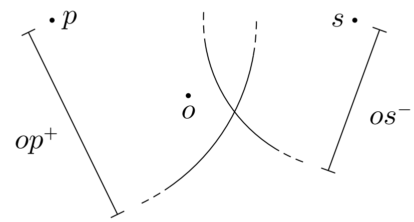
The applications of this theorem to pivot filtering have already been discussed in Sect. 4. However, its metric analogue Theorem 2 is also used for overlap checking with balls and shells, which is what the upper and lower limits to the distances represent. The notion of overlapping metric balls is inherently triangular, and does not directly translate to Ptolemaic distances. We are still able to exploit similar information, but we are in the somewhat unusual situation of working with two balls (or shells) at once. Take the upper bound, in the case where we know qp and qs (and, of course, ps). In order to apply this bound (for automatic inclusion of o), we would have to know that o falls inside two balls, one around p and one around s, with radii op^+ and os^+, respectively. While this sort of "double containment" is not the norm in current metric indexing methods, it is certainly not impossible to implement. One would simply need to let each region be represented by two distance balls, and maintain two covering radii—one for each center. There is an inherent tradeoff here: If the objects are far apart, the covering radii will necessarily become quite large; however, if we move them closer, the upper bound will increase.

In considering the *lower* bound, we see something interesting: If we envision a structure with covering radii around both p and s, the lower limits op^- and os^- both become zero, leaving us also with a total lower bound of zero. We see that, as for the pivot filtering case, we may need for one of the pivots to be more "query-like," and the pivots need to be different from each other. For example, if the query is close to p but far from s, and the converse holds for the object, the lower bound will be high.

However, this may not be enough. Consider the case where the object falls within a ball with radius r around p (giving us $os^- = ps - r$); we then get the following variation of the lower bound from (2):



(a) intersecting shells



(b) outer and inner radius

Figure 5: Two intersecting shells of a PM-Tree or some similar structure. Both shells contain the objects of the region, se we are free to use the upper radius of one and the lower of the other for added Ptolemaic filtering.

$$qo \geq \frac{1}{ps} \cdot (qp \cdot (ps - r) - r \cdot qs) = qp - r \cdot \frac{qp + qs}{ps}$$

The only difference from the triangular condition is the scaling factor $(qp + qs)/ps$, which we can see as regulating the influence of the radius. If the query lies directly between p and s (that is, $ps = qp + qs$) this new bound is, in fact, exactly equivalent to the triangular one. However, for all other cases, the new bound is *worse*.

What's missing is the "other half," as it were: an *inverted* ball around s, excluding o, giving us a proper os^-, in addition to the covering radius, op^+. This would be available in a situation where we have covering shells from each pivot to its sibling regions (as in GNAT [35] and its descendants, or the PM-Tree, which uses subtree shells around global pivots [36]), or where we use "inside/outside" partitioning with multiple pivots simultaneously (as in D-Index [37]). In such cases, where both upper and lower bounds are available for both op and os, the bound in Theorem 3 can be used directly, substituting exact values for ps, qs and qp.[7] See Fig. 5 for an example.

Finally, for Ptolemaic *metrics*, it is possible to combine the upper or lower bounds generated by Ptolemaic pivoting with various metric regions. This is similar to the technique used in such metric index structures as the CM-Tree [38] and TLAESA [39, 40], where a lower bound to the center object is used in ball overlap checking, and in Hybrid Dynamic SA-Tree [41], where the same approach is taken in hyperplane filtering. The idea is to be able to filter out an entire region without computing the distance to its representative objects (such as centers or vantage points). For example, for a ball with center c and covering radius r_c, as well as a query radius r_q the normal discard criterion would be $r_q < qc - r_c$. Instead, with the pivots p and s, we could replace qc with a Ptolemaic lower bound (see Fig. 6). The criterion then becomes

$$r_q < \frac{qp \cdot cs - qs \cdot cp}{ps} - r_c \,.$$

For a hyperplane defined by the two centers u and v, the region corresponding to v can

[7]This, in fact, is the technique used in implementing the Ptolemaic PM-Tree [12].

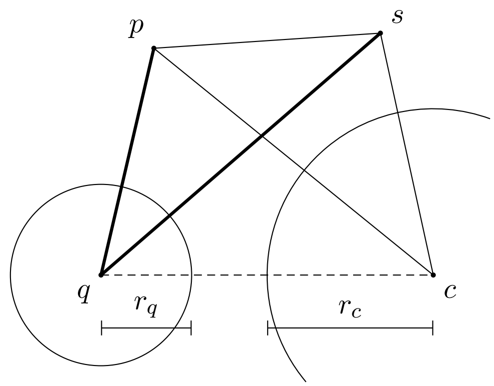


Figure 6: Computing pivot distances (heavy) at the start of the search potentially lets us eliminate regions without examining their centers. The available distances (solid) combine to form a lower bound on qc; if this bound is greater than $r_q + r_c$, overlap is impossible.

normally be excluded if $2r_q < qv - qu$. Again, we can avoid calculating the distance qv by using a lower bound in its place, with the criterion

$$2r_q < \frac{qp \cdot vs - qs \cdot vp}{ps} - qu\,.$$

In both cases, multiple pivots would, of course, strengthen the bounds. Similar techniques could also be used with metric shells, or even the vantage point regions of the VP-Tree [42, 43] and the D-Index [37].

6 Experimental Results

Two sets of experiments have been performed to explore the potential usefulness of Ptolemy's inequality in distance indexing: The first set of experiments evaluate its filtering power (using pivot filtering), while the second evaluates its recall rates for non-Ptolemaic metrics, for possible use in approximate search.

6.1 Filtering Power for Ptolemaic Metrics

The first set of experiments were designed to compare the filtering power of the Ptolemaic and triangular approaches. Fig. 7 shows the results for pivot filtering, with both the triangular and Ptolemaic lower bounds.

The first data set consisted of uniformly random 10-dimensional vectors, and the queries used for evaluation were drawn from the same distribution. For the second set, 20-dimensional data and query vectors were drawn in equal proportion from ten gaussian clusters (with $\sigma^2 = 0.1$), centered around vectors drawn uniformly random from the unit hypercube. This clustering approach is similar to that used by Zezula et al. [30]. The last data set consisted of 64-dimensional image histograms under a quadratic form distance similar to that described by Hafner et al. [17], using Euclidean distance in RGB space as the basis for the weight matrix.[8] The histograms themselves were constructed by posterizing random images from an online image repository [44]. For the first two data sets, the data set consisted of 100 000 vectors, with 20 and 50 pivots, respectively. For the last data set, the 100 queries and 20 pivots were sampled without replacement from the original set of 10 000 images, and removed before the searches were performed.

The pivot filtering was performed directly on a precomputed distance table in the style of LAESA [26]. In addition to the full Ptolemaic pivot filtering described in Sect. 4, a limited version was tested with only the $n-1$ consecutive pairs of pivots, in an arbitrary ordering, giving $2n-2$ bounds of the form of (7), because the absolute value was used. This gives a CPU use closer to that of traditional pivoting. Of course, there is a range of settings available here, from a linear number of bounds, to the quadratic number used by the full filter. The plots in Fig. 7 show the total number distance computations needed beyond the number of pivots (i.e., the number of objects not eliminated). The search radius (the horizontal axis) is described using the number of objects it encompasses beyond the query (10–50). In each case, m is the number of pivots used. The results were averaged over 100 random queries. As can be seen, for these cases the full Ptolemaic approach clearly outperforms the triangular, with the partial Ptolemaic filtering also offering an advantage of varying significance.

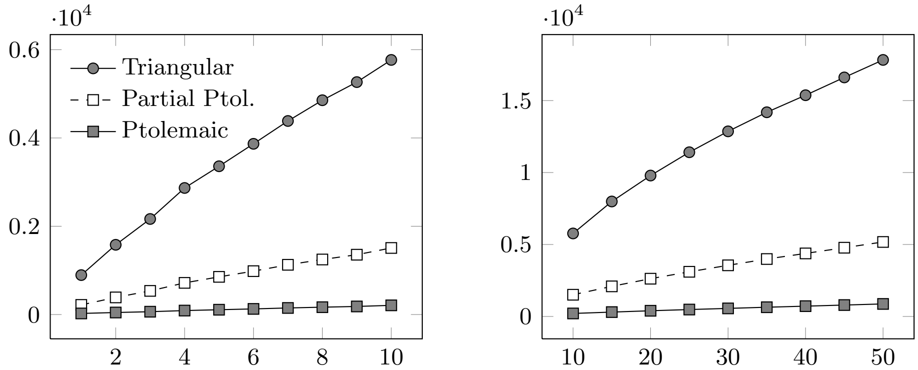
It is worth noting that the Ptolemaic filters used in Fig. 7 are *pure*, that is, not combined with a triangular bound. Fig. 8 illustrates the relative contribution of the triangular and Ptolemaic pivoting bounds, if both are used in range search where the radius covers the ten nearest objects, on 100 000 uniformly random ten-dimensional vectors, averaged over 100 queries. Queries and pivots were drawn from the same distribution as the data. The filtering power is measured as the number of objects eliminated by the given bounds during search. The four regions represent (*1*) objects filtered out only by the Ptolemaic bound, (*2*) objects that both bounds are able to filter out, (*3*) objects filtered out only by the triangular bound, and (*4*) objects that neither bound manages to disqualify. A progression similar to that in Fig. 8 was also found for clustered data, with a somewhat less pronounced difference (data not shown).

Even for a relatively modest number of pivots, the filtering power of the Ptolemaic bound is high, and the difference in the exclusive contributions of the two bounds is surprisingly large, growing to the extreme for higher dimensions. This can perhaps be seen even more clearly in Fig. 9, where the individual (non-exclusive) filtering powers of the triangular, partial Ptolemaic and full Ptolemaic methods are plotted independently.

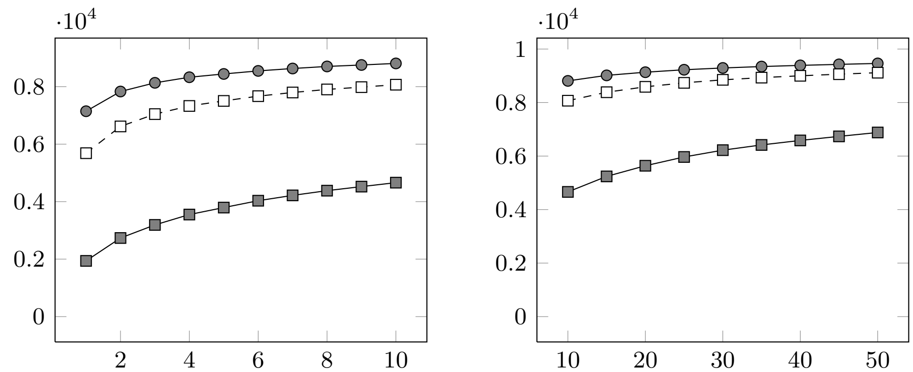
6.2 Approximation Rates for Non-Ptolemaic Metrics

Although I focus on exact search in this paper, the ideas may be applicable to approximate search as well, and could perhaps also be used with merely *approximately* Ptolemaic distances. While any metric can be made Ptolemaic, as mentioned in Sect. 2, the resulting

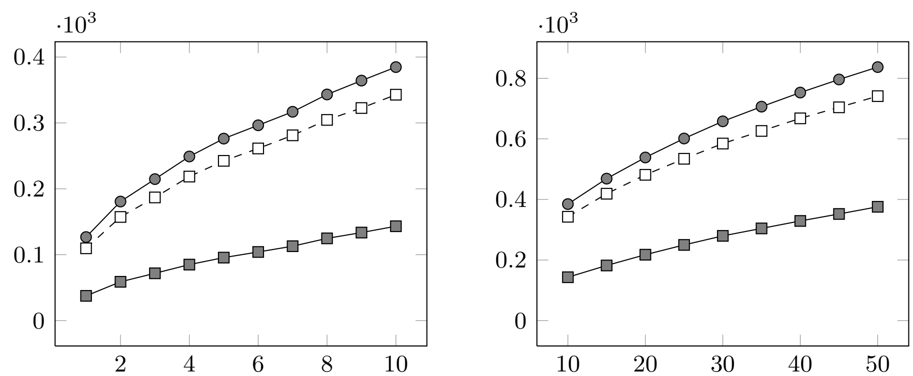
[8]The matrix used was $a_{ij} = 1 - d_{ij}/d_{\max}$, where i and j are histogram bins.



(a) L_2, uniform, $\mathbb{R}^{10}$, $m = 20$, $n = 100\,000$



(b) L_2, clustered, $\mathbb{R}^{20}$, $m = 50$, $n = 100\,000$



(c) QFD, image histograms, $\mathbb{R}^{64}$, $m = 20$, $n = 9880$

Figure 7: Distance computations (y-axis) for range search with m pivots over n objects, radii covering 1–50 neighbors (x-axis), averaged over 100 random queries. The clustered data consisted of 10 Gaussian clusters, each with $\sigma^2 = 0.1$. The queries for the uniform case were uniform. For the clusters, 10 queries were generated from each cluster distribution; for the image histograms, the pivots and queries were sampled without replacement, and removed from the original 10 000 vectors.

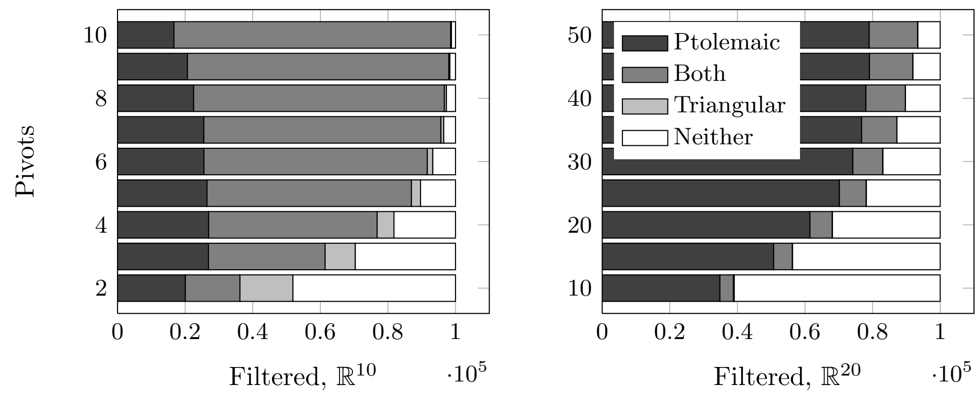


Figure 8: Filtering contributions on uniformly random vectors, for the given number of random pivots. The shaded areas represent the number of objects that were filtered exclusively by the triangular and Ptolemaic bounds, by both, or by neither.

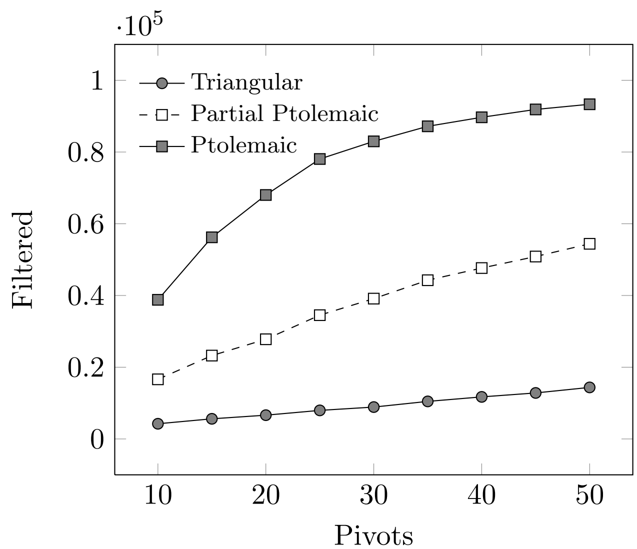


Figure 9: Individual filtering power for uniformly random vectors from $\mathbb{R}^{20}$. The y-axis is the number of objects filtered by each of the three bounds individually, independently of the other two, for the given number of random pivots.

metric may be harder to index, and approximate indexing of the original may still be useful. Table 1 shows results for several data sets and distances. The three types of objects used were vectors, sets and strings. The vectors were generated as described in Sect. 6.1. The sets were generated randomly for a given maximum cardinality by including or excluding each object with equal probability. The string data sets were a list of 234 936 words from Webster's Second International Edition, as distributed with the Macintosh OS version 10.5, and the lines of *A Tale of Two Cities* by Charles Dickens, with short lines (fewer than six characters) stripped away. The latter data set is similar to one used by Brin [35]. For the vector data sets, various L_p norms were used, as well as the angular pseudometric. For the sets, the related Hamming and Jaccard distances were used (the cardinality of the symmetric difference, and proportion of the symmetric difference to the union, respectively), while for the strings, Levenshtein distance (edit distance) was used. Each experiment was averaged over ten runs. Except for the string data sets, which were static, new data sets were randomly generated for each run. A run consisted of 10 000 randomly sampled quadruples (that is, sets of four distinct objects), and the proportion of the quadruples that satisfied Ptolemy's inequality was computed.

As can be seen from Table 1, the much-used L_p norms vary in their approximation rates from 99% to 100% in the higher-dimensional spaces (with L_2, of course having an exact rate of 100%). The angular pseudometric does very well for the high-dimensional vectors. The set distances also seem to conform to Ptolemy's inequality to a high degree for the 20-dimensional case, and the edit distance has a very low rate of inequality violations (with the Dickens data set outperforming the Webster data set by several orders of magnitude). For the L_p spaces and the angular pseudometric, the intrinsic dimensionality was generally close to the actual number of dimensions (slightly higher for uniformly random vectors, and varying with p). The dimensionalities for the Hamming spaces were equal to about half the set cardinalities, while those for Jaccard distance were about 50% greater. The intrinsic dimensionality of the Webster data set was about nine, while that for the Dickens data set was about forty.

7 Discussion and Future Work

In summary, this paper has presented a new distance indexing principle, based on Ptolemy's inequality, which seems to result in greatly increased filtering power, in the cases where it is applicable. For vector spaces (in particular, $\mathbb{R}^k$), it is directly applicable to quadratic form distances, including Euclidean distance. The cost of the filtering depends on the number of bounds used for a given number of pivots, growing from linear, as with triangular filtering, up to quadratic, providing a spectrum of options which could form the basis of a tradeoff between filtering cost and the performance gained by eliminating distance computations. The exact tradeoff will of course depend on the computational cost of the distance function. The quadratic form distance is, at the face of it, an excellent candidate in this regard, as it is quite expensive to compute. However, static quadratic form distances can be normalized to Euclidean distance, transforming the query vector prior to search. This leaves dynamic versions, such as the signature quadratic form distance, as an important application. Results published after the original version of this paper shows that Ptolemaic indexing is, indeed,

Table 1: Results from approximation experiments. The values are average Ptolemaic proportions of quadruples, as well as standard deviations. Subscripts on numbers represent powers of ten. $\mathbb{R}^k$ refers to vectors of dimension k, while $2^{\mathbb{Z}_k}$ are sets of cardinality k.

Distance		Clustered, $\mathbb{R}^5$		Clustered, $\mathbb{R}^{10}$		Uniform, $\mathbb{R}^{10}$	
		μ	σ	μ	σ	μ	σ
L_1	Manhattan distance	0.98	1.70_{-3}	1.00	3.81_{-4}	1.00	8.63_{-4}
L_2	Euclidean distance*	1.00	—	1.00	—	1.00	—
L_3		0.99	6.39_{-4}	1.00	1.11_{-4}	1.00	4.58_{-5}
L_5		0.98	1.41_{-3}	1.00	6.03_{-4}	1.00	2.14_{-4}
L_{10}		0.97	1.39_{-3}	0.99	1.31_{-3}	1.00	4.94_{-4}
L_{100}		0.96	2.01_{-3}	0.98	1.49_{-3}	1.00	5.27_{-4}
L_∞	Chebyshev distance	0.96	2.09_{-3}	0.98	1.45_{-3}	1.00	5.48_{-4}
θ	Angular distance	0.99	5.78_{-4}	1.00	1.19_{-4}	1.00	4.00_{-5}
		Uniform, $2^{\mathbb{Z}_{10}}$		Uniform, $2^{\mathbb{Z}_{20}}$			
Hamming distance		0.93	2.69_{-3}	0.98	9.57_{-4}		
Jaccard distance		0.99	8.90_{-4}	1.00	1.60_{-4}		
		Webster		Dickens			
Levenshtein distance		1.00	7.81_{-5}	1.00	0.00		

* Euclidean distance is Ptolemaic, but is included for completeness

currently the most efficient way to index these distances [11, 12]. Beyond such empirical validation in a more realistic setting, there are other avenues of research that might be interesting to pursue. For example, as shown, there also several common distances that are *close* to being Ptolemaic. Whether Ptolemaic indexing of such spaces might turn out to be useful as an approximate indexing method could be an issue worth investigating further. It might also be interesting to examine the basic properties of the Ptolemaic pivots, as related to the space (including which composition of pivots that would give the best pivoting results). Some basic engineering work in this direction has already met with experimental success, such as exploring pivot pairs in order of distance from the query and the object in question, to achieve elimination more quickly [11, 12].

It seems that higher dimensionalities give Ptolemaic indexing a greater edge over the triangular one. In the experiments performed here, this may be due to the increased number of pivots, but even the partial Ptolemaic filtering increases its lead with with higher dimension and a higher pivot number, which cannot be explained by the quadratic growth in the number of bounds. This is certainly an issue worthy of further examination. It also seems like high-dimensional spaces are more likely to be approximately Ptolemaic—perhaps because the distances are more similar. Based on this reasoning, high-dimensional spaces would also be more likely to be approximately triangular. Taking the square root, which will increase the intrinsic dimension, will make any metric *exactly* Ptolemaic. On the other hand, both the Ptolemaic and the triangular bound will be weaker in these cases, so there is clearly a tradeoff between accuracy and filtering power. These are issues that could be examined both empirically and analytically.

Acknowledgements

The author wishes to thank Ole Edsberg, Jon Marius Venstad and Bilegsaikhan Naidan for their help in several areas in the preparation of this paper, as well as the anonymous reviewers for their helpful suggestions.

References

[1] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[2] Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis. *R-Trees : Theory and Applications*. Advanced Information and Knowledge Processing. Springer, 2006.

[3] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.

[4] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[5] G. R. Hjaltason and H. Samet. Index-driven similarity search in metric spaces. *ACM Transactions on Database Systems, TODS*, 28(4):517–580, December 2003.

[6] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search : The Metric Space Approach*. Springer, 2006.

[7] M. L. Hetland. The basic principles of metric indexing. In C. A. C. Coello, S. Dehuri, and S. Ghosh, editors, *Swarm Intelligence for Multi-objective Problems in Data Mining*, volume 242 of *Studies in Computational Intelligence*. Springer, 2009.

[8] A. W. Fu, P. M. Chan, Y. Cheung, and Y. S. Moon. Dynamic VP-tree indexing for n-nearest neighbor search given pair-wise distances. *The VLDB Journal*, 9(2):154–173, July 2000.

[9] T. Bartoš, T. Skopal, and J. Moško. Towards efficient indexing of arbitrary similarity: Vision paper. *ACM SIGMOD Record*, 42(2):5–10, 2013.

[10] T. Bartoš and T. Skopal. Designing similarity indexes with parallel genetic programming. In *Similarity Search and Applications*, pages 294–299. Springer, 2013.

[11] J. Lokoč, M. L. Hetland, T. Skopal, and C. Beecks. Ptolemaic indexing of the signature quadratic form distance. In *Proceedings of the Fourth International Conference on Similarity Search and Applications*, pages 9–16. ACM, 2011.

[12] M. L. Hetland, T. Skopal, J. Lokoč, and C. Beecks. Ptolemaic access methods: Challenging the reign of the metric space model. *Information Systems*, 38(7):989–1006, 2013.

[13] W. A. Wilson. On certain types of continuous transformations of metric spaces. *American Journal of Mathematics*, pages 62–68, 1935.

[14] I. J. Schoenberg. On metric arcs of vanishing menger curvature. *Annals of Mathematics*, 41(4):715–726, October 1940.

[15] E. Deza and M. M. Deza. *Dictionary of Distances*. Elsevier, 2006.

[16] J. D. Smith. Generalization of the triangle and ptolemy inequalities. *Geometriae Dedicata*, 50(3):251–259, 1994.

[17] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[18] T. Bernas, E. K. Asem, J. P. Robinson, and B. Rajwa. Quadratic form: A robust metric for quantitative comparison of flow cytometric histograms. *Cytometry Part A*, 73A(8):715–726, 2008.

[19] I. J. Schoenberg. A remark on M. M. Day's characterization of inner-product spaces and a conjecture of L. M. Blumenthal. *Proceedings of the American Mathematical Society*, 3(6):961–964, December 1952.

[20] T. Skopal, T. Bartoš, and J. Lokoč. On (not) indexing quadratic form distance by metric access methods. In *Proceedings of the 14th International Conference on Extending Database Technology*, pages 249–258. ACM, 2011.

[21] C. Beecks, M. S. Uysal, and T. Seidl. Signature quadratic form distance. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 438–445. ACM, 2010.

[22] B. J. Malešević. The Möbius–Pompeïu metric property. *Journal of Inequalities and Applications*, 2006(83206):1–9, 2006.

[23] T. Foertsch and V. Schroeder. Hyperbolicity, CAT(−1)-spaces and the Ptolemy inequality. arXiv:math.MG/0605418 v2 13 July, 2006.

[24] T. Skopal. Unified framework for exact and approximate search in dissimilarity spaces. *ACM Transactions on Database Systems, TODS*, 32(4), 2007.

[25] C. D. Feustel and L. G. Shapiro. The nearest neighbor problem in an abstract metric space. *Pattern Recognition Letters*, 1(2):125–128, December 1982.

[26] M. L. Micó and J. Oncina. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recognition Letters*, 15(1):9–17, January 1994.

[27] O. Pedreira and N. R. Brisaboa. Spatial selection of sparse pivots for similarity search in metric spaces. In *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science*, number 4362 in Lecture Notes in Computer Science, pages 434–445, 2007.

[28] E. Vidal Ruiz. An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters*, 4(3):145–157, 1986.

[29] K. Figueroa, E. Chávez, G. Navarro, and R. Paredes. On the least cost for proximity searching in metric spaces. In C. Àlvarez and M. Serna, editors, *Proceedings of the 5th International Workshop on Experimental Algorithms*, volume 4007 of *Lecture Notes in Computer Science*, pages 279–290. Springer, 2006.

[30] P. Zezula, P. Ciaccia, and F. Rabitti. M-tree: A dynamic index for similarity queries in multimedia databases. Technical Report 7, HERMES ESPRIT LTR Project, 1996.

[31] V. Dohnal. An access structure for similarity search in metric spaces. In W. Lindner, M. Mesiti, C. Türker, Y. Tzitzikas, and A. Vakali, editors, *EDBT Workshops*, volume 3268 of *Lecture Notes In Computer Science*, pages 133–143. Springer, 2004.

[32] K. Fredriksson. Engineering efficient metric indexes. *Pattern Recognition Letters*, 2006.

[33] N. Brisaboa, O. Pedreira, D. Seco, R. Solar, and R. Uribe. Clustering-based similarity search in metric spaces with sparse spatial centers. In *Proceedings of the 34th Conference on Current Trends in Theory and Practice of Computer Science*, number 4910 in Lecture Notes in Computer Science, pages 186–197. Springer, 2008.

[34] M. L. Hetland. A survey of recent methods for efficient retrieval of similar time sequences. In M. Last, A. Kandel, and H. Bunke, editors, *Data Mining in Time Series Databases*, chapter 2, pages 23–42. World Scientific, 2004.

[35] S. Brin. Near neighbor search in large metric spaces. In U. Dayal, P. M. D. Gray, and S. Nishio, editors, *Proceedings of 21th International Conference on Very Large Data Bases, VLDB*, pages 574–584. Morgan Kaufmann, 1995.

[36] T. Skopal. Pivoting M-tree: A metric access method for efficient similarity search. In V. Snášel, J. Pokorn'y, and K. Richta, editors, *Proceedings of the Annual International Workshop on DAtabases, TExts, Specifications and Objects (DATESO)*, volume 98 of *CEUR Workshop Proceedings*, Desna, Czech Republic, April 2004. Technical University of Aachen (RWTH).

[37] V. Dohnal, C. Gennaro, P. Savino, and P. Zezula. D-index: Distance searching index for metric data sets. *Multimedia Tools and Applications*, 21(1):9–33, 2003.

[38] L. Aronovich and I. Spiegler. Efficient similarity search in metric databases using the CM-tree. *Data & Knowledge Engineering*, 2007.

[39] M. L. Micó, J. Oncina, and R. C. Carrasco. A fast branch & bound nearest neighbour classifier in metric spaces. *Pattern Recognition Letters*, 17(7):731–739, June 1996.

[40] K. Tokoro, K. Yamaguchi, and S. Masuda. Improvements of TLAESA nearest neighbour search algorithm and extension to approximation search. In *Proceedings of the 29th Australasian Computer Science Conference*, pages 77–83. Australian Computer Society, Inc., 2006.

[41] D. Arroyuelo, G. Navarro, and N. Reyes. Fully dynamic and memory-adaptative spatial approximation trees. In *Proceedings of the Argentinian Congress in Computer Science, CACIC*, pages 1503–1513, 2003.

[42] J. K. Uhlmann. Metric trees. *Applied Mathematics Letters*, 4(5):61–62, 1991.

[43] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 311–321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[44] Yahoo! Inc. *Flickr*, 2009. Available from http://flickr.com.